An Intelligent Vision-only Operator Interface for Dexterous Robots

Paolo Fiorini, Gene Chalfant
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Yuichi Tsumaki
Mechanical Engineering
Tohuku University
Sendai, Japan

Enrico Di Bernardo, Pietro Perona
Electrical Engineering
California Institute of Technology
Pasadena, California 91125

Abstract: *This paper describes the development of a vision-only intelligent interface for dexterous robots capable of tracking operator motions, analyzing trajectory feasibility, and providing visual feedback to the operator. This system analyses the operator's arm motion for safety, and then converts it into trajectory commands for a mechanical arm. Potentially dangerous situations, such as singularity and self collision, are displayed to the operator as graphical icons superimposed over the robot video images. The paper summarizes the main features of the current implementation, presents the approach developed for singularity identification and characterizes the performance of the demonstration prototype.*

1 Introduction

The goal of this project is to develop methods and algorithms enabling the natural and unencumbered generation of commands for anthropomorphic robots while minimizing specific operator training and dedicated equipment. The project aims at allowing the use of sophisticated, human-like robots in critical environments, such as the International Space Station and the proposed Human Settlement on Mars, where time, space and mass constraints prevent the use of more traditional telepresence systems, such as joysticks and force-feedback devices. In particular, this project focuses on two main areas: (1) *Visual tracking* of the operator body, to develop new machine vision algorithms enabling the identification and robust tracking of operator motions; and (2) *Intelligent Advisor*, to develop monitoring functions enabling the verification of the operator's commands and providing visual display of sensory and operations data. This work is motivated by the parallel development, at NASA Johnson Space Center, of the anthropomorphic robotic system Robonaut, pursuing the design of a robot the size of a suited astronaut. This robot will be teleoperated by a non-intrusive telepresence interface, taking advantage of the natural physical correspondence between the robot and the human operator.

Advanced teleoperator interfaces are characterized by several distinct elements. The input system used here is the visual tracking of the operator arm motions. In the past, visual tracking has been addressed with two main approaches: marker-based and marker-less systems. Marker-based systems [4, 5] use reflective markers on the body of the subject, infrared lighting and multiple cameras. This approach can compute the marker positions in 3D with very high accuracy (1 part over 3000 of the calibrated volume). The marker-less approach is based on modeling the appearance of body parts in the images generated by one or more cameras. In [6], the authors use an image intensity difference extracted from the real image and assume orthographic camera projections. The sum of the chamfer distances between each edges of the expected silhouette and the closest edge detected in the current image is used in [7]. The vector of distances between corresponding points on the occlusion boundaries of the real and the expected silhouette in each image plane is used in [8]. In [9, 10], the authors use direct measures on the background-subtracted image in the locations defined by the predicted body pose and the 3D body model.

Graphical displays have been a constant feature of teleoperation interfaces. In [11], a computer-generated graphical simulation is used to give the operator status informations and to compensate for transmission time delay. Intelligent interaction with the operator is addressed in in [13], where an interactive path planner for the teleoperation of dexterous manipulators is proposed. In [14], a program monitors task execution and gives the operator feedback about task status.

Kinematic understanding is critical for the correct teleoperation of a dexterous arm, since singularity location is not self-evident. In the past, singularity detection has been addressed primarily from the point of view of avoiding any unexpected and dangerous motion of the robot. In particular, approaches such as [15] solve the problem by forcing the robot to pass at a distance from the nominal trajectory, whereas

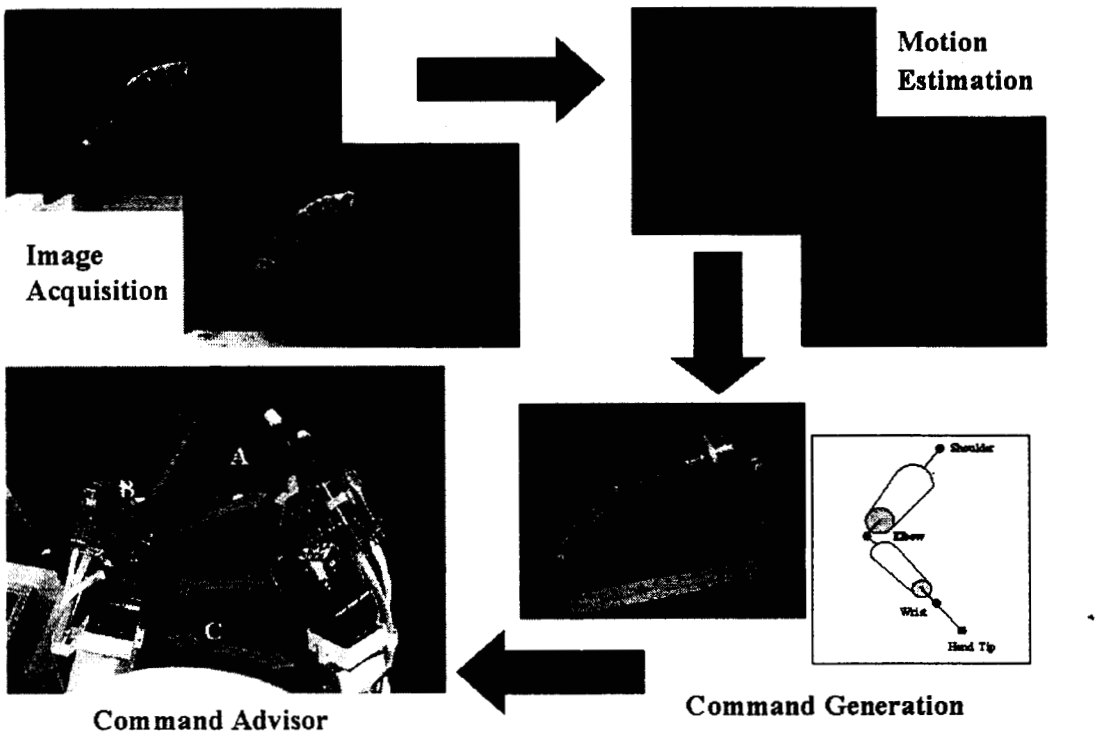

Figure 1: Conceptual organization of the intelligent interface.

solutions such as [17] keep the robot velocity within acceptable limits. In a teleoperation system these changes can confuse the operator, when they happen without a warning. Therefore, the proximity of a singularity area must be monitored and the operator warned.

The system presented here attempts to overcome the difficulties of teleoperating a dexterous manipulator by arm motion tracking, by displaying as virtual obstacles the kinematic difficulties , such as singularities and self-collision, associated to a given configuration of the remote manipulator, so that the operator can avoid these obstacles.

The rest of the paper describes the overall architecture of this interface, and its main components. Section 2 is a brief description of the data and action flow of this system. Section 3 describes the features of the visual tracker. Section 5 describes the self-collision detection. Section 6 describes the approach used to visualize the singularity areas relative to a given arm configuration. Finally, the last Section summarizes the main results of this work and describes the future development.

2 System Overview

The features of the proposed vision-based telepresence interface are demonstrated in an off-line system based on the graphical simulation of a Robotics Research Corp. (RRC) dexterous manipulator. The system described here implements the conceptual flow of data and actions represented schematically in Fig. 1. The action sequence stars at the top left of the Figure, with the operator moving his arm. The visual tracker then extracts the Cartesian coordinates of the critical points of the arm, i.e. shoulder, elbow and wrist. These coordinates are then converted into a set of joint angles for the remote manipulator, which is analyzed to identify possible self-collision situations and to compute the distance from kinematic singularities. The result of this analysis is then displayed to

the operator as a set of graphical icons superimposed on the real time video images of the remote site, as shown schematically at the bottom left of Fig. 1. The intelligence of the system is in the ability to analyze the robot commands and to inform the operator of unexpected situations, thus simplifying the operator understanding of the remote robot kinematic configuration.

Because of the heavy requirements that real-time performance puts on computation, this demonstrator operates off-line, i.e. each element of the system is a stand alone unit, and data are processed in batch-mode rather than in a real-time pipeline. However, the graphical display integrates all the various elements, providing a realistic simulation of the real-time performance.

3 The Visual Tracker

The hardware and software system performing the visual tracking of the operator motions consists of the following subsystems: front-end, acquisition hardware, the off-line calibration software, the 3D reconstructing software, and the labeling software.

The front-end for the measurement of the operator's arm motion is a non-invasive system based on a set of 4 video-cameras and small markers to be placed on the main joints of the operator's body. The current implementation simplifies the problem of detecting the markers in each camera, using small light bulbs as markers and requiring a low level of illumination in the acquisition area. The acquisition, detection and 3D reconstruction of the markers coordinates is carried out at a 60Hz frame rate and with an accuracy of 10 mm for a calibration volume of about 3 m by 3m.

Figure 2 shows the main hardware components of the acquisition system. We used 4 monochrome NTSC interlaced cameras with 6mm lenses located on a semi circle around the center of the calibrated volume at a distance of about 5 m from the center and about 3 m above the floor level.

On each camera the automatic gain was disabled and the shutter time was set to 1/500 sec as a trade off between image contrast and motion blur. One camera operates as a master, and an external trigger guarantees the synchronous start of the frame grabbers. The real-time process running on the host detects the intensity local maxima and saves the coordinates of those points on disk. This operation guarantees a 30Hz frame rate with no MPEG compression. Marker coordinate detection consists of (1) image blurring with a 5x5 Gaussian kernel 5x5 to low-pass filter local brightness maxima; (2) localize all pixels having

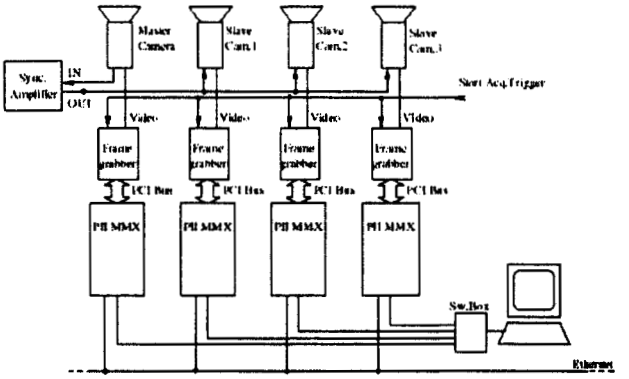

Figure 2: Block Diagram of the Acquisition System.

local maximum brightness; (3) brightness maxima detection with sub-pixel accuracy, up to 1/20 of pixel size. This phase produces a set of 4 files containing the coordinates of all detected markers in each image field.

Camera calibration consists of the estimation of: internal parameters, mutual camera position, and rigid transformation between cameras and world reference frame. The 3D reconstruction software triangulates between different views to determine the 3 dimensional position of each markers. This operation is complicated by the fact that correspondences between markers in different channels are not known and some markers might not even be present in some channels due to occlusions. Therefore it is necessary to efficiently parse through the set of possible marker correspondences and find the best hypothesis at each time frame. We found that robustness to image noise and the computational cost are improved by using temporal segments of marker coordinates of variable length which are built by tracking each marker on the image plane. The tracking is done in a very conservative way, interrupting the thread as soon as the marker disappears or gets too close to another marker.

This visual tracking algorithm takes the set of reconstructed 3D marker coordinates at each frame and assign to each one the correct original body part. The program requires to manually label the 3D points in one of the frames in the sequence. It then uses threads and probability information to propagate the labeling to the adjacent frames. This data constitute the end

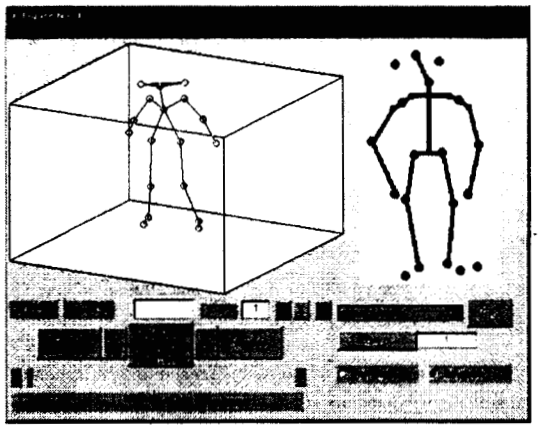

Figure 3: The Visualization/Editing Tool.

product of the visual front-end and are provided to the system that controls the robotic arm.

The visualization tool shown in Figure 3 was implemented in Matlab to provide a graphical representation of the computed data (for example representing the reconstructed body with a stick figure), for inspection and editing.

4 The Graphical Display

The system described here interacts with the operator exclusively via a graphical interface which incorporates live video of the robot workspace. A graphic model of the robot, with a viewpoint matching the video camera, is overlaid on the video. Robot joint angles computed using the visual tracker data control the configuration of the graphic model. Various indicators can be placed in the scene; these currently include self-collision warnings and visualization of the singularity volumes, as described later. Figure 4 shows the graphical simulation as a transparent image superimposed to the real time video of the remote site. To indicate self-collisions, it appears useful to highlight the two links which are in danger of colliding. Alternatively, a line can be drawn connecting the points of closest approach. Singularity zones in Cartesian space are irregular volumes which change as the joint angles change. Visualizing these presents a number of problems such as determining the best simplified geometric representation, visibility and viewpoint considerations, determining an adequate resolution (trading off computation time), and promoting the most effective intuition of the workspace constraints. Our current implementation displays a singularity volume as an assemblage of cubes of different color which represent

Figure 4: The overlay graphical simulation.

the proximity of the robot end-effector to a singularity.

5 Self-Collision Detection

Obstacle detection has been used in other operator interfaces for space robotics. In particular, we use the approach developed in [16], briefly summarized here for completeness. Self collision is defined as the situation when a manipulator link is too close to one of its other links. In this case, it is sufficient to test for line-segment to line-segment distances, taking into account the radii of the links. When a manipulator link is approaching collision with a link face, for example terminal plate or motor sides, then the nearest point to the link on the face will either be (1) the nearest point to the link along one of the edges of the face, or (2) the nearest point of the distal end-point of the link to the face. When the distal end-point is the nearest point to the face, then the projection of the distal end-point into the face plane will be within the boundaries of the face.

For link-to-link distances, the minimum distance between the link axis line segments is computed and the radii of both links are subtracted from this distance in order to get the minimum distance between link surfaces. The geometric link surface that this

models is a cylinder of uniform radius with hemispherical ends of the same radius.

For link-to-edge distances, the minimum distance between the link axis line segment and the edge line segment is computed and the link radius is subtracted in order to get the minimum link-to-edge distance. The nearest points are the points on the link axis line segment and the edge line segment that are closest to each other.

For link-to-face distances, the distance between the distal end-point of the link and the plane of the face is computed, as well as the projection of the end-point in the plane. If the projection of the distal end-point is not within the face, then this end-point-to-face measurement is discarded.

6 Visualization of Singularity Space

The system described here uses as the teleoperator slave an RRC dexterous manipulator, whose kinematics is studied in [1]. The approach used to generate the joint angles and resolve the kinematic redundancy, given the set of end-effector positions generated by the visual tracker, is the Configuration Control method [15]. In particular, redundancy is eliminated by assigning a known value to the *arm angle*, i.e. to the angle formed by the plane through the manipulator wrist, elbow and shoulder, and a reference plane containing the base z-axis and the robot wrist. Therefore, the arm angle can be computed from the operator elbow position. This condition adds algorithmic singularities to the arm own kinematic singularities. Since it is very difficult to identify these singular configurations, an operator may move the arm near a singularity without recognizing it.

Although configuration control prevents dangerous motions, it also alters the operator motion tracking, thus possibly confusing the operator and creating problems in tight environments. To solve this problem we generate graphical warnings to localize nearby singularities by using the distance from a singularity expressed as $\det(JJ^T)$ [2]. Since the augmented Jacobian J_A is used for the RRC dexterous manipulator, $\det(J_A J_A^T)$ expresses the distance from both kinematic and algorithmic singularities. We visualize the singularity volume in the vicinity of the manipulator wrist based on the value of the determinant.

With this approach, the space around the end-effector of the manipulator is uniformly discretized into cubes. We consider the center of each cube as the representative point of the cube. If $\det(J_A J_A^T)$ at this point becomes smaller than a preset threshold, the cube is visualized in red. When the manipulator

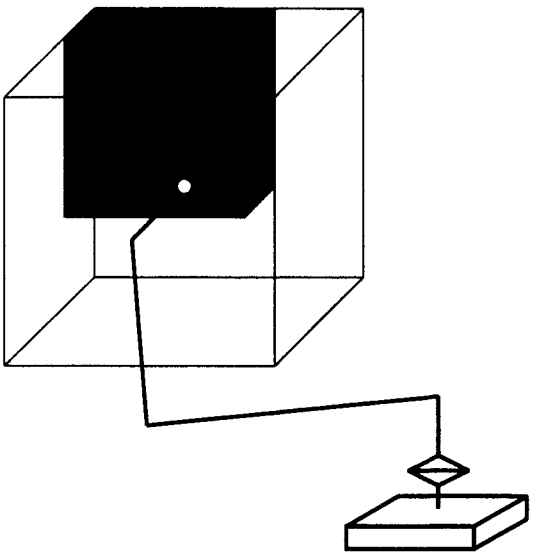

Figure 5: An example of singularity visualization.

is not tracking the reference trajectory in the vicinity of a singularity, we use the tracking error as a measure of the distance between the current manipulator position and the singularity. Points with this characteristic are visualized by a blue cube. Figure 5 shows schematically the two different representations.

In this system, the singularity space is 7-dimensional and therefore of difficult visualization and of even more difficult understanding. For this reason, we only visualize the 3-DOF singularity space relative to translational motion with fixed orientation and arm angle. However, since the shape of this 3-dimensional space depends also on the end-effector orientation and on the arm angle, the computed distance also expresses the relation to the complete singularity space generated including orientation and arm plane angle.

The computational cost of this approach depends primarily on the number of the cubes displayed. From our experience, it takes several seconds to calculate and visualize 13x13x13 points forming a search over a cubic volume of 520mm of side. Figure 6 shows the singularity space as a 13x13x13 tessellation. By reducing the resolution of the tessellation to a 3x3x3 representation, it is possible to calculate and visualize the singularity volume in real time. Furthermore, the singularity representation does not need to be recomputed if the end-effector does not move beyond the search volume and keeps the same orientation and arm angle. Of course, this representation is too coarse to understand the details of the singularity shape, but

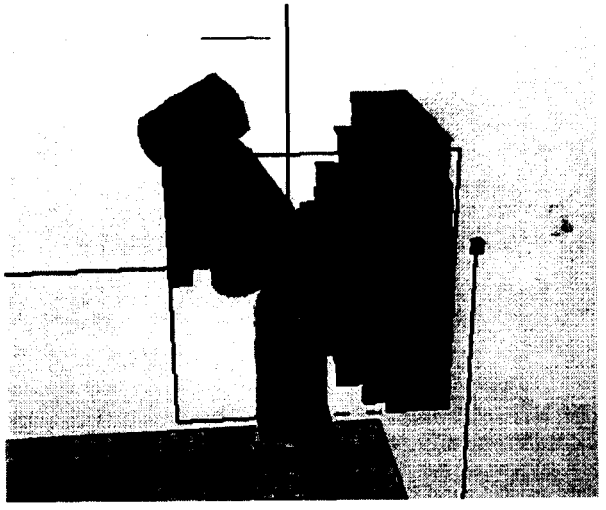

Figure 6: Singularity visualization with 13x13x13 cubes.

it is still useful to warn about and to avoid the singularity. Figure 7 shows an example of singularity representation using 3x3x3 cubes. Finally, the capability of changing the observation viewpoint is an important tool to understand the shape of the singularity volume. Therefore, we include in the interface the viewpoint changing features described in [3]. Figure 8 shows the singularity volume of Figure 6 from a different viewpoint.

7 Conclusions and Future Work

This paper describes a demonstration system designed to show the main features of a vision-only teleoperation interface for dexterous manipulators. The innovative features of this system include a visual tracker to teleoperate a remote manipulator by simply moving the arms, and an intelligent advisor, to analyze the commanded motions for kinematic correctness, enabling the operator to avoid self collision and singularities. Naturally, this work is only the beginning of the development of a completely vision-based operator interface, and several new features need to be added to this basic system. In particular, visual tracking technologies needs to be enhanced in: real-time computation of the 3D upper-body pose of the operator; system robustness with respect to body and objects occlusion, changes in the background, and clothing differences.

Similarly, the graphical representation needs to be tested with human subjects to identify the best combinations of graphical icons for an immediate and unambiguous understanding of the robot kinematic conditions. Other possible improvements are, for example, the computation of optimal trajectories from the operator motion, the visualization of annotations attached to task elements, the integration of additional video displays, and the use of the interface as a planning tool.

Acknowledgments: The research described in this paper has been carried out in part at the Jet Propulsion Laboratory, California Institute of Tech-

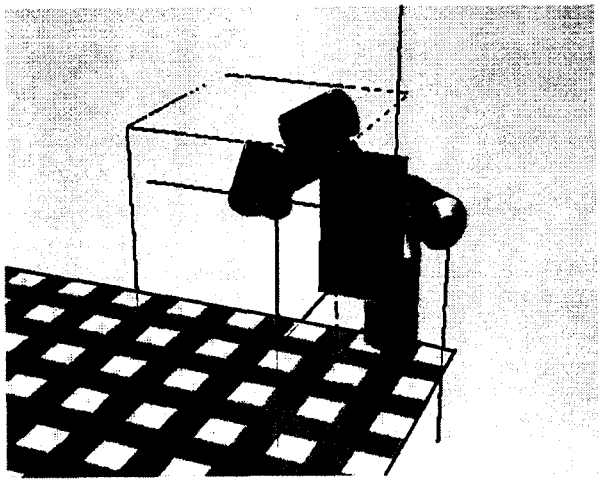

Figure 7: Singularity visualization with 3x3x3 cubes.

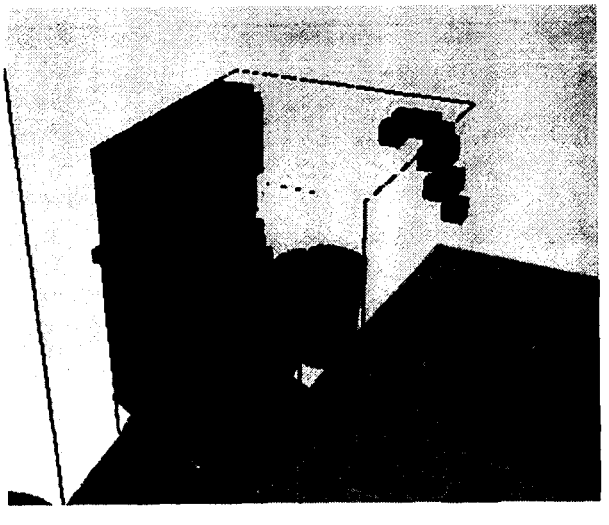

Figure 8: Singularity volume from another viewpoint.

nology, under contract with the National Aeronautics and Space Administration. Dr. Tsumaki research at JPL is supported by a grant from the Ministry of Education, Science, Sports and Culture of Japan (MONBUSHO).

References

[1] K. Kreutz-Delgado, M. Long, H. Seraji, "Kinematic Analysis of 7-DOF Manipulator," *The Int. J. of Robotics Research*, Vol. 11, No. 5, pp. 469–481, 1992.

[2] T. Yoshikawa, "Manipulability of robotic mechanisms," *Robotics Research: The Second Int. Symp.* , H. Hanafusa and H. Inoue, eds, Cambridge, MA: MIT Press, pp. 440–446, 1984.

[3] Y. Tsumaki, S. Kotera, D. N. Nenchev, M. Uchiyama, "Advanced Experiments with a Teleoperation System Based on the SC Approach," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1196–1201, 1992.

[4] Web site: http://www.bts.it

[5] Web site: http://www.motionanalysis.com

[6] J.M. Rehg and T. Kanade, "DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction", *Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies*, November 1994, pp.16-24.

[7] D.M. Gavrila and L.S. Davis, "3-D Model-Based Tracking of Humans in Action: a Multi-View Approach", *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition*, June 1996, pp. 73-80, San Francisco CA.

[8] I.A. Kakadiaris and D. Metaxas, "Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection", *IEEE Conf. Comput. Vision and Pattern Recog.*, 6/96, pp. 81-87.

[9] L. Goncalves and E. Di Bernardo and E. Ursella and P. Perona, "Monocular Tracking of the human arm in 3D", *Proc. 5th Int. Conf. Computer Vision*, June 1995, pp. 764-770, Boston MA.

[10] E. Di Bernardo and L. Goncalves and P. Perona, "Monocular Tracking of the Human Arm: Real-Time Implementation and Experiments", *Proc. 13th Int. Conf. Pattern Recognition*, August 1996, pp. 622-626, Wien.

[11] P. Fiorini, A. Bejczy and P. Schenker, "Integrated Interface for Advanced Teleoperation", *IEEE Control System Magazine*, Vol. 13, N. 5, October 1993.

[12] S. Hayati, J. Balaram, H. Seraji, W.S. Kim, K.S. Tso, "Remote Surface Inspection System", *Robotics and Autonomous Systems*, 1993 11(1), pp. 45-49.

[13] P. Fiorini, H. Das, and S. Lee, "Man-Machine Cooperation in Advanced Teleoperation", *Proc. 6th Annual Space Operation, Application and Research Symposium* (SOAR'92), Houston, TX, August 4-6 1992.

[14] P. Fiorini, A. Giancaspro, S. Losito and G. Pasquariello, "Neural Networks for the Segmentation of Teleoperation Tasks", *Presence, Teleoperators and Virtual Environments*, Vol. 2, N.1, pp. 1-13, MIT Press, 1993.

[15] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Trans. on Robotics and Automation*, 5(4):473-490.

[16] B. Bon and H. Seraji, "Real-time Model-based Obstacle Detection for the NASA Ranger Telerobot", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1997.

[17] Y. Tsumaki, D. Nenchev, S. Kotera, and M. Uchiyama, "Teleoperation Based on the Adjoint Jacobian Approach", *IEEE Control System Magazine*, pp. 53-62, February 1997.

[18] H. Ding and W. Schiehlen: "On controlling robots with redundancy in an environment with obstacles," Proc. Symp. on Robot Control, pp. 771-776, Capri, Italy, 1994.

[19] J. Cohen, M. Lin, D. Manocha and K. Ponamgi: "I-COLLIDE: an interactive and exact collision detection system for large-scaled environments," Proc. ACM Int. 3D Graphics Conference, pp. 189-196, 1995.

[20] E. Cheung and V. Lumelsky: "Proximity sensing in robot manipulator motion planning: system and implementation issues," IEEE Trans. on Robotics and Automation, pp. 740-751, 1989.

[21] J. Feddema and J. Novak: "Whole arm obstacle avoidance for teleoperated robots," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 3303-3309, San Diego, 1994.

[22] H. Seraji, R. Steele and R. Ivlev: "Sensor-based collision avoidance: Theory and experiments," Journal of Robotic Systems, vol. 13, no. 9, pp. 571-586, 1996.